NAME

libunibetacode − Library for Beta Code to Unicode conversion

SYNOPSIS

int
ub_beta2greek (*char *beta_string*, *int max_beta_string*, *char *utf8_string*, *int max_utf8_string*);

int
ub_beta2coptic (*char *beta_string*, *int max_beta_string*, *char *utf8_string*, *int max_utf8_string*);

int
ub_beta2hebrew (*char *beta_string*, *int max_beta_string*, *char *utf8_string*, *int max_utf8_string*);

int
ub_greek2beta (*char *utf8_string*, *int max_utf8_string*, *char *beta_string*, *int max_beta_string*);

int
ub_coptic2beta (*char *utf8_string*, *int max_utf8_string*, *char *beta_string*, *int max_beta_string*);

int
ub_hebrew2beta (*char *utf8_string*, *int max_utf8_string*, *char *beta_string*, *int max_beta_string*);

int
ub_codept2utf8 (*unsigned codept*, *char *utf8_bytes*);

int
ub_utf82codept (*char *utf8_bytes*, *unsigned codept*);

DESCRIPTION

libunibetacode is a self-contained C library with functions to convert between UTF-8 Unicode and Beta Code, as adopted by the University of California, Irvine Thesaurus Linguae Graecae (TLG) Program and the Tufts University Perseus Project, among others. Beta Code provides a way of encoding polytonic Greek characters using plain ASCII characters. Beta Code also provides some support for encoding Coptic and Hebrew.

The **libunibetacode** package contains three top-level functions to convert from Beta Code to UTF-8 Unicode, and three top-level functions to convert from UTF-8 Unicode to Beta Code.

The top-level functions to convert Beta Code to UTF-8 Unicode are:

> **ub_beta2greek**(3) converts a Greek Beta Code input string to a UTF-8 output string.
>
> **ub_beta2coptic**(3) converts a Coptic Beta Code input string to a UTF-8 output string.
>
> **ub_beta2hebrew**(3) converts a Hebrew Beta Code input string to a UTF-8 output string.

The top-level functions to convert UTF-8 Unicode to Beta code are:

> **ub_greek2beta**(3) converts a Greek UTF-8 input string to a Greek Beta Code output string.
>
> **ub_coptic2beta**(3) converts a Coptic UTF-8 input string to a Coptic Beta Code output string.
>
> **ub_hebrew2beta**(3) converts a Hebrew UTF-8 input string to a Hebrew Beta Code output string.

In addition:

> **ub_codept2utf8**(3) converts a Unicode *code point* to a UTF-8 output string.
>
> **ub_utf82codept**(3) converts a Unicode UTF-8 string to to a Unicode code point.

A Unicode code point is an assignment to a specific numeric value for glyphs and other entities in Unicode fonts. By convention, Unicode code points are given by their Unicode numeric values in the form U+xxxx, where "xxxx" is a string of four hexadecimal digits representing a glyph in the Unicode Basic Multilingual Plane.

All of these functions are non-destructive: they will not alter the input strings that are passed to them.

State is not preserved between calls to any of these functions.

The Beta Code conversion functions (**ub_beta2greek**, **ub_beta2coptic**, and **ub_beta2hebrew**) expect the input string to contain only Beta Code sequences for Greek, Coptic, or Hebrew, respectively. Likewise, the language-specific UTF-8 to Beta Code conversion functions (**ub_greek2beta**, **ub_coptic2beta**, and **ub_hebrew2beta**) expect the input string to contain only UTF-8 code points that map to valid Beta Code sequences in the respective language.

The functions **ub_codept2utf8** and **ub_utf82codept** support the entire Unicode space of U+0000 through U+10FFF. Thus they are not tied to one Beta Code language (Greek, Coptic, or Hebrew), and so can complement the other functions.

libunibetacode supports the language-specific Beta Code letter and punctutation symbol mappings described in **unibetacode**(5).

The additional capabilities described in **unibetacode**(5) section "EXTENSIONS FOR ASCII AND UNICODE" are not implemented. There is also no function to perform the equivalent of the standalone program **unibetaprep**(1). As a consequence, **ub_beta2greek** does not support the full Beta Code numeric sequence range beginning with ’#’ and followed by a decimal number. For example, the Unicode Byzantine Music Symbols having TLG Beta Code encodings of ’#2000’ through ’#2245’ (corresponding to Unicode code points U+1D000 through U+1D0F5) are not supported.

The three Beta Code to UTF-8 Unicode functions also do not support the Unicode code point description format of the form "\uxxxx" that **beta2uni**(1) supports. That limits the usefulness of **ub_beta2hebrew**, because the TLG Beta Code specification only contains encodings for Hebrew consonants, not for vowels or cantillation marks. A user program could use **ub_codept2utf8** along with **ub_beta2hebrew** to fill this gap.

Balanced double quotes are supported in **ub_beta2greek** and **ub_beta2coptic**, but the opening and closing quotation marks must appear in the same input string because state is not preserved between calls to those functions. (An input string can contain embedded newlines.) Quotation marks in **ub_beta2hebrew** are output as the ASCII double quote mark (").

The **ub_greek2beta** function will determine whether a Greek letter follows a lower-case sigma in the input UTF-8 string, and based upon that convert Greek medial and final small sigma to "s" if context will make the conversion back from Beta Code to UTF-8 unambiguous. If this is not the case, small sigma will be converted to "s1" for small medial sigma or "s2" for small final sigma. For example, if a final sigma is followed by a letter, then the final sigma will be converted to Beta Code as "s2" to ensure proper conversion back from Beta Code into UTF-8.

Note: Thesaurus Linguae Graecae and TLG are registered trademarks of the University of California.

PARAMETERS

The top-level functions described in this document take the following parameters:

beta_string A null-terminated string with Beta Code sequences for the corresponding script (Greek, Coptic, or Hebrew). This string is an input for functions that convert from Beta Code to UTF-8, and an output for functions that convert from UTF-8 to Beta Code.

max_beta_string The maximum size of *beta_string*, in bytes, to prevent accesses past the end of the array.

utf8_string A null-terminated string with UTF-8 Unicode sequences for the corresponding script (Greek, Coptic, or Hebrew). This string is an output for functions that convert from Beta Code to UTF-8, and an input for functions that convert from UTF-8 to Beta Code.

max_utf8_string The maximum size of *utf8_string*, in bytes, to prevent accesses past the end of the array.

codept An unsigned 32-bit Unicode code point. This is an input to **ub_codept2utf8**, and an output from **ub_utf82codept**.

utf8_bytes The null-terminated UTF-8 byte string corresponding to the Unicode code point stored in *codept*. This is an output from **ub_codept2utf8**, and an input to **ub_utf82codept**.

UNICODE GREEK

The Greek Extended range of The Unicode Standard (U+1F00 − U+1FFF) contains 16 small and capital vowels that have identical representation in the Greek and Coptic range (U+0370 − U+03FF). These are vowels with an *oxia* (acute) accent in the Greek Extended range; they have equivalent glyphs with a *tonos* (acute) accent in the Greek and Coptic range. Because of this duplication, the use of these 16 Greek Extended glyphs is deprecated.

However, unlike the **beta2uni** program, by default the function **ub_beta2greek** maps to those 16 deprecated code points. This was done after observing that many fonts contain consistent looking glyphs in the Unicode Greek Extended block that do not have a consistent appearance with the Greek and Coptic block.

The choice between these two options is compiled in with a #define statement near the beginning of "`ub_beta2greek.c`", which is in the "src/libsrc" directory in the source distribution. To avoid conversion to these 16 deprecated code points, change the following two lines:

```
// #define GREEK_COMBINING beta2combining
#define GREEK_COMBINING beta2combining_alt
```

to this:

```
#define GREEK_COMBINING beta2combining
// #define GREEK_COMBINING beta2combining_alt
```

and then recompile the package by running "`make`" in the top-level package source directory.

RETURN VALUES

Each of the library functions returns the number of bytes in the UTF-8 output string, not including the final null character that terminates the string.

SAMPLES

The directory "examples" in the source distribution contains samples with mappings from Beta Code to UTF-8 and vice versa. The "genesis-1-1.beta" and "genesis-1-1.utf8" files show the Bible verse Genesis 1:1 in Koine Greek (from the Septuagint), Hebrew, and Bohairic Coptic in Beta Code and UTF-8, respectively.

The program "test/ublibcheck.c" in the source distribution is a sample program that calls **ub_beta2greek**, **ub_beta2coptic**, and **ub_beta2hebrew** to convert the above-mentioned Genesis 1:1 passage from Beta Code to UTF-8. Then **ub_greek2beta**, **ub_coptic2beta**, and **ub_hebrew2beta** are invoked for a round-trip conversion. If the round-trip conversion from Beta Code to UTF-8 and back to Beta Code succeeds for all three languages, the program exits with an exit status of 0. Each of the three Beta Code to UTF-8 functions calls **ub_codept2utf8** to produce its UTF-8 output. Hence this program tests all of the top-level library functions. Once the "`make install`" command above has completed, the test program can be copied to another directory and compiled separately as a starting point for new software as follows:

```
cc ublibcheck.c -o ublibcheck -lunibetacode
```

SEE ALSO

unibetaprep(1), **beta2uni**(1), **uni2beta**(1), **unibetacode**(5)

AUTHOR

The **unibetacode** package was created by Paul Hardy.

LICENSE

BUGS

No known bugs exist. However, all corner cases have not been tested.